UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/547,288	04/11/2000	Nir N. Shavit	33226/959002; P4663	4871

32615 7590 02/22/2008
OSHA LIANG L.L.P./SUN
1221 MCKINNEY, SUITE 2800
HOUSTON, TX 77010

EXAMINER	
LI, AIMEE J	
ART UNIT	**PAPER NUMBER**
2183	
NOTIFICATION DATE	**DELIVERY MODE**
02/22/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

lord@oshaliang.com
hernandez@oshaliang.com
DOCKETING@OSHALIANG.COM

Office Action Summary	Application No.	Applicant(s)	
	09/547,288	SHAVIT ET AL.	
	Examiner	Art Unit	
	AIMEE J. LI	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 27 November 2007.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 40,41 and 43-56 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 40,41 and 43-56 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 11 April 2000 is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date 4/13/2004.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 40-41 and 43-56 have been considered. Claims 40-41, 43, 46-48, 52-53, and 56 have been amended as per Applicants' request.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as filed 27 November 2007.

Information Disclosure Statement

3. The information disclosure statement (IDS) submitted on 13 April 2004 is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the examiner.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claims 40, 43, 47, and 52 rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The claims, taking claim 40 as exemplary, contain language such as "wherein elements may be added to and removed..." and "...an atomic dual target compare and swap (DCAS) operation operable to update...". This language is unclear whether the system actually performs these operations or has these characteristics. The language insinuates that the system is capable of or may be able to perform these operations or have these characteristics, instead of clearly and distinctly stating the claims will perform these operations or have these characteristics. The Examiner suggests changing the language to "wherein the

elements are added to and removed…" and "…an atomic dual target compare and swap (DCAS) operation updates…" or similar to show that clear intent of the claim language.

6. Claim 52 recites the limitation "the first DCAS operation" in the last line of the claim. There is insufficient antecedent basis for this limitation in the claim. No previous "first DCA operation" has been established.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 40-41 and 43-56 are rejected under 35 U.S.C. 103(a) as being unpatentable over Janice M. Stone's "A simple and correct shared-queue algorithm using Compare-and-Swap" ©1990 in view of Kruse, Leung, and Tondo's Data Structures and Program Design in C ©1991 (herein referred to as Kruse) and in further view of Mark Allen Weiss's Data Structures & Algorithm Analysis in C++ ©1999.

9. Referring to claim 40, Stone has taught a computer program product encoded in at least one computer readable medium, the computer program product comprising:

a. Instances of the at least one function sequence concurrently executable by plural processors of a multiprocessor and each comprising an atomic dual target compare and swap (DCAS) operation operable to update an element of the array and one of the first end identifying index and the second end identifying index (Stone Section **1. Introduction**; Section **2. Background**, paragraphs 3 and 5-6;

Section **3. Compare-and-Swap**; Section **3.1 The Compare-and-Swap Instruction**, paragraph 1; Section **3.2 The A-B-A problem**, paragraphs 2-3; Section 3.3 **The Compare-and-Swap-Double Instruction**, paragraphs 1 and 4; Section **4. The Shared-Queue Algorithm**; Section **5. State diagram for the shared queue**, paragraphs 1-2 and 5; and Figure 5); and

b. Wherein the DCAS of the at least one functional sequence is responsive to a corresponding boundary condition state of the concurrent shared object (Stone Section **1. Introduction**; Section **2. Background**, paragraphs 3 and 5-6; Section **3. Compare-and-Swap**; Section **3.1 The Compare-and-Swap Instruction**, paragraph 1; Section **3.2 The A-B-A problem**, paragraphs 2-3; Section 3.3 **The Compare-and-Swap-Double Instruction**, paragraphs 1 and 4; Section **4. The Shared-Queue Algorithm**; Section **5. State diagram for the shared queue**, paragraphs 1-2 and 5; and Figure 5).

10. Stone has not taught at least one function sequence implementing an access operation on a concurrent shared double-ended queue (deque), wherein the deque is implemented as a circular buffer on a contiguous array of bounded size, wherein the deque is delimited by a first end identifying index and a second end identifying index, wherein the first end identifying index identifies a first element of the array adjacent to a first end element of the deque and the second end identifying index identifies a second element of the array adjacent to a second end element of the deque. Kruse has taught at least one function sequence implementing an access operation on a concurrent shared double-ended queue (deque), wherein the deque is implemented as a circular buffer on a contiguous array of bounded size, wherein the deque is delimited by a first

end identifying index and a second end identifying index, wherein the first end identifying index identifies a first element of the array adjacent to a first end element of the deque and the second end identifying index identifies a second element of the array adjacent to a second end element of the deque (Kruse pages 69-72). In regards to Kruse, the circular queue has a beginning location at 0 and ending location at max-1, which denote the beginning and ending locations of the bounded array in memory. The circular queue array also has a front and rear indices to denote the first and last locations of occupied locations in the array, and these locations are adjacent to the beginning and ending locations of the array. A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Kruse, that a circular queue array improves the efficiency of the array (Kruse page 69, section 3. Circular Arrays). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the circular buffer array of Kruse in the device of Stone to improve array efficiency.

11. In addition, Stone has not taught wherein elements may be added to and removed from both a first end and a second end of the deque. Weiss has taught wherein elements may be added to and removed from both a first end and a second end of the deque (Weiss page 118, problem 3.28 and page 492, problem 11.13). A person of ordinary skill in the art at the time the invention was made would have recognized that a deque that has elements added and removed from both ends of the queue is more flexible than a standard queue. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the deque of Weiss in the device of Stone to increase structural flexibility.

12. Referring to claim 41, Stone in view of Kruse has taught

a. Wherein the at least one function sequence is at least one selected from a group consisting of a right pop operation, a left pop operation, a right push operation, and a left push operation (Stone Section **1. Introduction**; Section **2. Background**, paragraphs 3 and 5-6; Section **3. Compare-and-Swap**; Section **3.1 The Compare-and-Swap Instruction**, paragraph 1; Section **3.2 The A-B-A problem**, paragraphs 2-3; Section 3.3 **The Compare-and-Swap-Double Instruction**, paragraphs 1 and 4; Section **4. The Shared-Queue Algorithm**; Section **5. State diagram for the shared queue**, paragraphs 1-2 and 5; and Figure 5);

b. Wherein the boundary condition state corresponding to the right push operation and the left push operation is a full state of the deque (Kruse pages 69-72); and

c. Wherein the boundary condition state corresponding to the right pop operation and the left pop operation is an empty state of the deque (Stone Section **1. Introduction**; Section **2. Background**, paragraphs 3 and 5-6; Section **3. Compare-and-Swap**; Section **3.1 The Compare-and-Swap Instruction**, paragraph 1; Section **3.2 The A-B-A problem**, paragraphs 2-3; Section 3.3 **The Compare-and-Swap-Double Instruction**, paragraphs 1 and 4; Section **4. The Shared-Queue Algorithm**; Section **5. State diagram for the shared queue**, paragraphs 1-2 and 5; and Figure 5).

13. Referring to claim 43, Stone has taught an apparatus comprising:

a. Plural processors (Stone Abstract; Section **1. Introduction**; Section **2. Background**, paragraphs 3 and 5-6; Section **3. Compare-and-Swap**; Section **3.1 The Compare-and-Swap Instruction**, paragraph 1; Section **3.2 The A-B-A**

problem, paragraphs 2-3; Section 3.3 **The Compare-and-Swap-Double Instruction**, paragraphs 1 and 4; Section **4. The Shared-Queue Algorithm**; Section **5. State diagram for the shared queue**, paragraphs 1-2 and 5; and Figure 5);

b. A store addressable by each of the plural processors (Stone Abstract; Section **1. Introduction**; Section **2. Background**, paragraphs 3 and 5-6; Section **3. Compare-and-Swap**; Section **3.1 The Compare-and-Swap Instruction**, paragraph 1; Section **3.2 The A-B-A problem**, paragraphs 2-3; Section 3.3 **The Compare-and-Swap-Double Instruction**, paragraphs 1 and 4; Section **4. The Shared-Queue Algorithm**; Section **5. State diagram for the shared queue**, paragraphs 1-2 and 5; and Figure 5);

c. Means for coordinating competing access operations, the coordinating means employing in each instance thereof, at least one atomic dual target compare and swap (DCAS) operation to disambiguate a retry state and a boundary condition state of the deque based on then-current contents of an array element and one, but not both, of first- and second-end index stores (Stone Abstract; Section **1. Introduction**; Section **2. Background**, paragraphs 3 and 5-6; Section **3. Compare-and-Swap**; Section **3.1 The Compare-and-Swap Instruction**, paragraph 1; Section **3.2 The A-B-A problem**, paragraphs 2-3; Section 3.3 **The Compare-and-Swap-Double Instruction**, paragraphs 1 and 4; Section **4. The Shared-Queue Algorithm**; Section **5. State diagram for the shared queue**, paragraphs 1-2 and 5; and Figure 5).

14. Stone has not taught first- and second-end index stores accessible to each of the plural processors for identifying opposing ends of a double-ended queue (deque) encoded in circular buffer form in the addressable store, wherein the first- and second-end index stores identify elements of the array adjacent to the corresponding opposing ends of the deque. Kruse has first- and second-end index stores accessible to each of the plural processors for identifying opposing ends of a double-ended queue (deque) encoded in circular buffer form in the addressable store, wherein the first- and second-end index stores identify elements adjacent to the ends of the deque. (Kruse pages 69-72). In regards to Kruse, the circular queue has a beginning location at 0 and ending location at max-1, which denote the beginning and ending locations of the bounded array in memory. The circular queue array also has a front and rear indices to denote the first and last locations of occupied locations in the array, and these locations are adjacent to the beginning and ending locations of the array. A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Kruse, that a circular queue array improves the efficiency of the array (Kruse page 69, section 3. Circular Arrays). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the circular buffer array of Kruse in the device of Stone to improve array efficiency.

15. In addition, Stone has not taught wherein elements may be added to and removed from both a first end and a second end of the deque. Weiss has taught wherein elements may be added to and removed from both a first end and a second end of the deque (Weiss page 118, problem 3.28 and page 492, problem 11.13). A person of ordinary skill in the art at the time the invention was made would have recognized that a deque that has elements added and removed from both

ends of the queue is more flexible than a standard queue. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the deque of Weiss in the device of Stone to increase structural flexibility.

16. Referring to claim 44, Stone in view of Kruse has taught the apparatus of claim 43, wherein the access operations are selected from a group consisting of a right pop operation, a left pop operation, a right push operation, and a left push operation (Stone Section **1. Introduction**; Section **2. Background**, paragraphs 3 and 5-6; Section **3. Compare-and-Swap**; Section **3.1 The Compare-and-Swap Instruction**, paragraph 1; Section **3.2 The A-B-A problem**, paragraphs 2-3; Section 3.3 **The Compare-and-Swap-Double Instruction**, paragraphs 1 and 4; Section **4. The Shared-Queue Algorithm**; Section **5. State diagram for the shared queue**, paragraphs 1-2 and 5; and Figure 5).

17. Referring to claim 45, Stone in view of Kruse has taught the apparatus of claim 43, wherein each of the first- and second-end index stores identifies a next element in the array for adding a value to the deque (Kruse pages 69-75).

18. Referring to claim 46, Stone in view of Kruse has taught the computer program product of claim 40, wherein each of the pair of end-identifying indices identifies a next element in the array for adding a value to the deque (Kruse pages 69-75).

19. Referring to claim 47, Stone has taught a method of managing concurrent access to shared data, comprising performing a first pop operation on the deque, wherein a first atomic dual target compare and swap (DCAS) operation is executed using the first end identifying index and the end element to remove the end element from the deque (Stone Section **1. Introduction**; Section **2. Background**, paragraphs 3 and 5-6; Section **3. Compare-and-Swap**; Section **3.1 The**

Compare-and-Swap Instruction, paragraph 1; Section **3.2 The A-B-A problem**, paragraphs 2-3; Section 3.3 **The Compare-and-Swap-Double Instruction**, paragraphs 1 and 4; Section **4. The Shared-Queue Algorithm**; Section **5. State diagram for the shared queue**, paragraphs 1-2 and 5; and Figure 5).

20. Stone has not taught implementing a concurrent shared double-ended queue (deque) as a contiguous bounded-size array in a memory of a computer system, wherein a first end identifying index identifies a first array element adjacent to an end element of the deque and a second end identifying index identifies a second array element adjacent to the end element of the deque. Kruse has taught implementing a concurrent shared double-ended queue (deque) as a contiguous bounded-size array in a memory of a computer system, wherein a first end identifying index identifies a first array element adjacent to an end element of the deque and a second end identifying index identifies a second array element adjacent to the end element of the deque (Kruse pages 69-72). In regards to Kruse, the circular queue has a beginning location at 0 and ending location at max-1, which denote the beginning and ending locations of the bounded array in memory. The circular queue array also has a front and rear indices to denote the first and last locations of occupied locations in the array, and these locations are adjacent to the beginning and ending locations of the array. A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Kruse, that a circular queue array improves the efficiency of the array (Kruse page 69, section 3. Circular Arrays). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the circular buffer array of Kruse in the device of Stone to improve array efficiency.

21. In addition, Stone has not taught wherein elements may be added to and removed from both a first end and a second end of the deque. Weiss has taught wherein elements may be added to and removed from both a first end and a second end of the deque (Weiss page 118, problem 3.28 and page 492, problem 11.13). A person of ordinary skill in the art at the time the invention was made would have recognized that a deque that has elements added and removed from both ends of the queue is more flexible than a standard queue. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the deque of Weiss in the device of Stone to increase structural flexibility.

22. Referring to claim 48, Stone in view of Kruse has taught the method of claim 47, further comprising performing a second pop operation on the deque concurrently with the first pop operation, wherein a second DCAS operation is executed using the second end identifying index and the end element, wherein when the second DCAS operation fails, an indication of an empty state of the deque is returned from the second DCAS operation (Stone Section **1. Introduction**; Section **2. Background**, paragraphs 3 and 5-6; Section **3. Compare-and-Swap**; Section **3.1 The Compare-and-Swap Instruction**, paragraph 1; Section **3.2 The A-B-A problem**, paragraphs 2-3; Section 3.3 **The Compare-and-Swap-Double Instruction**, paragraphs 1 and 4; Section **4. The Shared-Queue Algorithm**; Section **5. State diagram for the shared queue**, paragraphs 1-2 and 5; and Figure 5).

23. Referring to claim 49, Stone in view of Kruse has taught the method of claim 47, wherein the deque is implemented as a circular buffer (Kruse pages 69-72).

24. Referring to claim 50, Stone in view of Kruse has taught the method of claim 48, wherein the first pop operation is performed by a first processor and the second pop operation is

performed by a second processor (Stone Abstract; Section **1. Introduction**; Section **2. Background**, paragraphs 3 and 5-6; Section **3. Compare-and-Swap**; Section **3.1 The Compare-and-Swap Instruction**, paragraph 1; Section **3.2 The A-B-A problem**, paragraphs 2-3; Section 3.3 **The Compare-and-Swap-Double Instruction**, paragraphs 1 and 4; Section **4. The Shared-Queue Algorithm**; Section **5. State diagram for the shared queue**, paragraphs 1-2 and 5; and Figure 5).

25. Referring to claim 51, Stone in view of Kruse has taught the method of claim 48, wherein the first pop operation is a left pop operation, the first end identifying index is a left-end index, and the first array element is to the left of the end element, and wherein the second pop operation is a right pop operation, the second end identifying index is a right-end index, and the second array element is to the right of the end element (Stone Section **1. Introduction**; Section **2. Background**, paragraphs 3 and 5-6; Section **3. Compare-and-Swap**; Section **3.1 The Compare-and-Swap Instruction**, paragraph 1; Section **3.2 The A-B-A problem**, paragraphs 2-3; Section 3.3 **The Compare-and-Swap-Double Instruction**, paragraphs 1 and 4; Section **4. The Shared-Queue Algorithm**; Section **5. State diagram for the shared queue**, paragraphs 1-2 and 5; and Figure 5).

26. Referring to claim 52, Stone has taught a method of managing concurrent access to shared data, comprising performing a first push operation on the deque, wherein a first atomic dual compare and swap (DCAS) operation is executed using the first end identifying index, the array element, and a value, wherein the value is stored in the array element as a result of the first DCAS operation (Stone Section **1. Introduction**; Section **2. Background**, paragraphs 3 and 5-6; Section **3. Compare-and-Swap**; Section **3.1 The Compare-and-Swap Instruction**, paragraph

1; Section **3.2 The A-B-A problem**, paragraphs 2-3; Section 3.3 **The Compare-and-Swap-Double Instruction**, paragraphs 1 and 4; Section **4. The Shared-Queue Algorithm**; Section **5. State diagram for the shared queue**, paragraphs 1-2 and 5; and Figure 5).

27. Stone has not taught implementing a concurrent shared double-ended queue (deque) as a contiguous bounded-size array in a memory of a computer system, wherein the deque comprises a first end element, a second end element, a first end identifying index, and a second end identifying index, and wherein the first end identifying index and the second end identifying index identify an array element, wherein the array element is adjacent to the first end element of the deque and the second end element of the deque. Kruse has taught implementing a concurrent shared double-ended queue (deque) as a contiguous bounded-size array in a memory of a computer system, wherein the deque comprises a first end element, a second end element, a first end identifying index, and a second end identifying index, and wherein the first end identifying index identifies an array element as adjacent to the first end element and the second end identifying index identifies the array element as adjacent to the second end element (Kruse pages 69-72). In regards to Kruse, the circular queue has a beginning location at 0 and ending location at max-1', which denote the beginning and ending locations of the bounded array in memory. The circular queue array also has a front and rear indices to denote the first and last locations of occupied locations in the array, and these locations are adjacent to the beginning and ending locations of the array. A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Kruse, that a circular queue array improves the efficiency of the array (Kruse page 69, section 3. Circular Arrays). Therefore, it would have

been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the circular buffer array of Kruse in the device of Stone to improve array efficiency.

28. In addition, Stone has not taught wherein elements may be added to and removed from both a first end and a second end of the deque. Weiss has taught wherein elements may be added to and removed from both a first end and a second end of the deque (Weiss page 118, problem 3.28 and page 492, problem 11.13). A person of ordinary skill in the art at the time the invention was made would have recognized that a deque that has elements added and removed from both ends of the queue is more flexible than a standard queue. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the deque of Weiss in the device of Stone to increase structural flexibility.

29. Referring to claim 53, Stone in view of Kruse has taught the method of claim 52, further comprising performing a second push operation on the deque concurrently with the first push operation, wherein a second DCAS is executed using the second end identifying index and the array element (Stone Section **1. Introduction**; Section **2. Background**, paragraphs 3 and 5-6; Section **3. Compare-and-Swap**; Section **3.1 The Compare-and-Swap Instruction**, paragraph 1; Section **3.2 The A-B-A problem**, paragraphs 2-3; Section 3.3 **The Compare-and-Swap-Double Instruction**, paragraphs 1 and 4; Section **4. The Shared-Queue Algorithm**; Section **5. State diagram for the shared queue**, paragraphs 1-2 and 5; and Figure 5), wherein when the DCAS fails, an indication of a full state of the deque is returned from the DCAS (Kruse pages 69-72).

30. Referring to claim 54, Stone in view of Kruse has taught The method of claim 52, wherein the deque is implemented as a circular buffer (Kruse pages 69-72).

31. Referring to claim 55, Stone in view of Kruse has taught the method of claim 53, wherein the first push operation is performed by a first processor and the second push operation is performed by a second processor (Stone Abstract; Section **1. Introduction**; Section **2. Background**, paragraphs 3 and 5-6; Section **3. Compare-and-Swap**; Section **3.1 The Compare-and-Swap Instruction**, paragraph 1; Section **3.2 The A-B-A problem**, paragraphs 2-3; Section 3.3 **The Compare-and-Swap-Double Instruction**, paragraphs 1 and 4; Section **4. The Shared-Queue Algorithm**; Section **5. State diagram for the shared queue**, paragraphs 1-2 and 5; and Figure 5).

32. Referring to claim 56, Stone in view of Kruse has taught the method of claim 53, wherein the first push operation is a left push operation, the first end identifying index is a left-end index, and the first end element is to the right of the array element, and wherein the second push operation is a right push operation, the second end identifying index is a right-end index, and the second end element is to the left of the array element (Stone Abstract; Section **1. Introduction**; Section **2. Background**, paragraphs 3 and 5-6; Section **3. Compare-and-Swap**; Section **3.1 The Compare-and-Swap Instruction**, paragraph 1; Section **3.2 The A-B-A problem**, paragraphs 2-3; Section 3.3 **The Compare-and-Swap-Double Instruction**, paragraphs 1 and 4; Section **4. The Shared-Queue Algorithm**; Section **5. State diagram for the shared queue**, paragraphs 1-2 and 5; and Figure 5).

Response to Arguments

33. Applicant's arguments with respect to claims 40-41 and 43-56 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

34. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. Leach et al., U.S. Patent Numbers 5,710,925; 5,745,764; 5,805,885; 6,240,465; and 6,243,764, have taught adding and removing elements from both ends of a list.

35. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

36. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

37. Any inquiry concerning this communication or earlier communications from the examiner should be directed to AIMEE J. LI whose telephone number is (571)272-4169. The examiner can normally be reached on M-T 7:00am-4:30pm.

38. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

39. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Aimee J Li/
Examiner, Art Unit 2183

13 February 2008

Aimee J. Li